Parallel-Chain Provenance with Structural Merge-Absence: A Cross-Domain Substrate for Multi-Authority Audit Trails

Authors: Berkan Karakus[1] (corresponding) ORCID iD: 0009-0004-0471-1069.

With domain attribution: NGS substrate (5th domain) identified by the author's mother; psychiatric assessment substrate (6th domain) identified in dialogue with Emre Büyükasık; bank credit intelligence substrate (8th domain, including the confidentiality-preserving cross-reference primitive) identified by the author's father (banker, VakıfBank, Türkiye).

[1] Independent researcher, Türkiye. Correspondence: berkankarakus21@gmail.com

Preprint version: 2026-05-14, Rev 1.5 working-draft (Zenodo v1.5 publication pending) **Status:** workshop-PROPOSED, pre-peer-review. **Zenodo concept DOI (always latest):** 10.5281/zenodo.20127704 **Zenodo v1.4 (Rev 1.4) DOI:** 10.5281/zenodo.20149539 — this version **Zenodo v1.2 (Rev 1.3) DOI:** 10.5281/zenodo.20130032 — prior version **Zenodo v1.1 (Rev 1) DOI:** 10.5281/zenodo.20128304 — prior version **Zenodo v1 (Rev 0) DOI:** 10.5281/zenodo.20127705 — original 2026-05-11 record **License:** CC BY 4.0 **Cite this version:** Karakus, B. (2026). *Parallel-Chain Provenance with Structural Merge-Absence: A Cross-Domain Substrate for Multi-Authority Audit Trails* (v1.4). Zenodo. https://doi.org/10.5281/zenodo.20149539 **Cite latest version:** Karakus, B. (2026). *Parallel-Chain Provenance with Structural Merge-Absence: A Cross-Domain Substrate for Multi-Authority Audit Trails*. Zenodo. https://doi.org/10.5281/zenodo.20127704

Rev 1 revision notes (vs. published Rev 0): Section 2 expanded with CRDT, transparency-log, secure-audit-log, database-provenance, named-graph, and privacy-preserving-record-linkage literature; archival "parallel provenance" terminology disambiguated; §7.2 formal-verification roadmap made concrete (TLA+); §7.3 cryptographic-enforcement roadmap made concrete (MPC, PSI, TEE); narrower "combination" novelty wording added. No substrate code or test count changes. Both Rev 0 (v1) and Rev 1 (v1.1) are independently citable Zenodo records and are linked by the same concept DOI; Rev 0 remains the original 2026-05-11 prior-art establishment. The post-v1.1 working draft in this file additionally incorporates editorial-pass corrections (§5.1 narrower combination wording carried through, "court-admissible" softened to "independently verifiable", Appendix-A absolutist language softened to "intended to", footer corrected to Rev 1, ORCID line punctuation).

Rev 1.3 working-draft notes (vs. Zenodo v1.1, not yet published as a Zenodo version): §2.8 tightened per a 2026-05-11 ninth-AI review (Elicit) which observed that the §5 `get_co_lender_existence → (bool, int)` signature *matches* the output shape of PSI-cardinality. The revised §2.8 paragraph now explicitly concedes this (PSI-cardinality is well-established and not claimed as novel), and locates the narrower defended claim in the *embedding* of PSI-cardinality semantics inside a non-fusible multi-authority chain substrate alongside identity-by-history and regulator-scoped metadata exclusion — i.e. the substrate-level composition. This sharpening pre-empts the obvious reviewer objection ("isn't this just PSI?") and hardens the preprint's positioning against a §101 Alice-doctrine eligibility challenge in any future narrow patent context. No substrate code or test count changes.

Rev 1.4 working-draft notes (vs. Rev 1.3, not yet published as a Zenodo version): Four ratified additions per Stream H disposition (`workshop/dispositions/main25_disposition_rev1.4_2026-05-12.md`): (1) §2.8 Para 2–3 replaced (Option B) with a substrate/API-layer framing paragraph (PSI-cardinality concession front-

loaded) plus four layer-correct contrast paragraphs against VeilAudit [32], GeoCross [33], zkCross [34], and ForensiCross [35] — sharpening the substrate-vs-application/protocol-layer distinction; (2) a new §1.3 paragraph after the Contributions list articulating qualitative wrapper-cost implications of substrate-level placement, with explicit overclaim disclaimer (no quantified reduction claim); (3) a new §4 intro sentence before §4.1 contrasting eight-domain breadth against the narrower anchor settings of the four cited cross-chain systems; (4) bibliography entries [32]–[35] appended after [31]. The zkCross [34] venue is the 33rd USENIX Security Symposium (USENIX Security '24), August 2024 — confirmed and corrected from any prior ACM-pattern DOI attribution. §2.8 Para 1 (PPRL/AudiWFlow/Harpocrates intro) and §6.3 (cross-architecture caveat) are intentionally untouched. No substrate code or test count changes. Published as Zenodo v1.4 (DOI 10.5281/zenodo.20149539) on 2026-05-13.

Rev 1.5 working-draft notes (vs. Rev 1.4, not yet published as a Zenodo version): §2.8 Para 1 expanded with Vos et al. (2024) [36] as named exemplar for the §5 primitive's containment-bit component, with party-of-origin suppression per Vos §4.3. Bibliography entry [36] appended after [35]. No other §2.8 changes — Paras 2–7 ([32]–[35] cross-chain contrast cluster, ratified in Rev 1.4 / Stream H disposition `workshop/dispositions/main25_disposition_rev1.4_2026-05-12.md`) remain unchanged. Cluster scope is effectively (a) Vos-only by structural-territory fit: Vos sits in PSI/PPRL-family territory ([25] cluster), structurally distinct from the cross-chain confidentiality territory of [32]–[35]. Wright 2025 (arXiv:2506.13246) was reviewed during the same Rev 1.5 prior-art audit and excluded from §2.8 on combined cross-family Q4=NO + Craig Wright UK High Court 2024 credibility grounds; workshop audit-trail entry at `workshop/dispositions/main25_disposition_wright_2025_exclusion_2026-05-14.md`. Cross-family verification: Vos 2024 returned Q2 = NOT COVERED and Q4 = NO unanimous 4/4 with full paper access (SciSpace + Elicit + Grok ×2 runs); Wright 2025 returned Q4 = NO 3/3 with full paper access (SciSpace + Elicit + Grok Fast), with the Gemini outlier (Q4 = YES) methodologically defeated by self-reported partial access plus Q2 misapplication plus Conclusion-rhetoric-as-system-model. No substrate code or test count changes.

Abstract

We present a data-structural substrate for tracking parallel independent assessments of the same external object such that (i) each authority's chain remains structurally non-fusible, (ii) cross-chain references can be expressed without content disclosure across authorities, and (iii) each chain's history is cryptographically verifiable. The substrate (named `provenance_engine_v2`) provides identity-by-history at the primitive level: two records with identical fold-sequences are substrate-identical, while two records with diverging histories become substrate-distinct. We demonstrate applicability across **eight independent domains** — patent prosecution, verifiable credentials, software bill-of-materials, AI model lineage, NGS variant interpretation, psychiatric assessment, credit rating actions, and bank credit intelligence — yielding **1322 passing test cases (1322/1322)** across the eight wrapper implementations. Seven of the eight domains exhibit *merge-absence* (a structural property formalized as §15.3): two authorities assessing the same object produce parallel records that cannot be fused at the substrate level; only the verifiable credentials domain exercises *operations-with-merge*, demonstrating the substrate's ability to surface scope-limited operations explicitly. The eighth domain (bank credit intelligence) introduces a novel **confidentiality-preserving cross-reference primitive** that supports existence queries across parallel chains without content disclosure.

We position this work as **defensive prior art** for the broader pattern, established by Bitcoin OpenTimestamps anchors and a Zenodo DOI on 2026-05-11. The substrate itself remains a structural-mathematical contribution; we

make no clinical, financial-advisory, or medical-device claim, and the domain-specific implementations are synthetic-fixture demonstrations unless otherwise noted.

Scope of the novelty claim. Provenance representation [10, 23], secure audit logging [20, 21], transparency logs [22], distributed-data convergence [19], software supply-chain attestation [11–13], and privacy-preserving record linkage / private set intersection [25] are all mature literatures. We do not claim to introduce any of them. The narrower contribution defended here is the **combination** of (i) substrate-level merge-absence as a primitive API property, (ii) identity-by-history at the substrate level, and (iii) regulator-scoped, content-excluding cross-reference queries embedded in non-fusible multi-authority chains — instantiated and unit-tested across eight independent domains. We further note that the phrase *parallel provenance* has prior use in archival description [26], where it denotes plural contextual narratives over records; the present paper instead studies *parallel-chain provenance* as an operational data-structural property.

Keywords: provenance tracking, parallel audit trails, merge-absence, multi-authority assessment, confidentiality-preserving cross-reference, data-structure pattern, cross-domain methodology

---

1. Introduction

1.1 Motivation — three illustrative scenarios

Scenario 1: clinical genomics. ClinVar records 30+ submissions per common BRCA1/BRCA2 variant from independent laboratories worldwide. Each lab's interpretation chain is publicly visible. When the same variant is reclassified by one lab (e.g., VUS → Likely-Pathogenic on new evidence), readers must be able to reconstruct: which lab said what when, what evidence supported each call, and whether earlier calls were revoked. Existing genomics tooling (Variant Effect Predictor, VEP; Sophia DDM; OncoKB) returns *current consensus* by default and loses the underlying lineage [1, 2].

Scenario 2: Turkish corporate banking. A Turkish firm has parallel credit relationships with VakıfBank, Akbank, İş Bankası, and Garanti BBVA. Each bank's internal credit-intelligence chain on this borrower is *legally private* under Bankacılık Kanunu Md. 73 [3]; banks cannot share intelligence content with one another. Yet for risk management across the banking system, a regulator (BDDK) must be able to audit "how many banks are exposed to this borrower" and "what is the distribution of risk grades across them" without seeing the content. To our knowledge, existing systems address parts of this problem — private matching, confidential audit logs, or workflow evidence — but do not combine API-level merge-absence with multi-authority identity-by-history and regulator-scoped, content-excluding cross-reference queries (see §2.8 for the adjacent literature this substrate is positioned against).

Scenario 3: AI model provenance. A foundation model is fine-tuned by Lab A, distilled by Lab B, quantized by Lab C, and merged (LoRA / SLERP / TIES) by Lab D. Each transformation produces a **new model with a new history**; the original foundation model's substrate identity is unchanged. Existing model registries (HuggingFace Hub, MLflow) record metadata but do not enforce the structural property that a merged model is a NEW lineage, not a fusion of input lineages [4]. This matters when investigating training-data provenance disputes (e.g., NYT v. OpenAI).

1.2 Shared structural problem

All three scenarios share a single structural problem:

> ***Multiple independent authorities make parallel assessments of the same external object; we need to record what each authority said when, support revisions and revocations, prevent unauthorized fusion of independent chains, and enable verifiable queries — all without requiring a single "ground truth" authority.***

This problem appears across at least 8 distinct application domains (Section 4). We claim that a single substrate-level data structure solves it for all of them.

1.3 Contributions

1. **A substrate** (`provenance_engine_v2`) providing identity-by-history (E8), gauge-baseline σ-attestation (E26), and orbit-decomposition verification (E19) — primitives sufficient to express parallel-chain provenance natively (Section 3).
2. **A taxonomy of frictions** (F1–F5) that arise when mapping the substrate to domain-layer record-and-classify problems, with resolutions validated across 8 domains (Section 4).
3. **A merge-absence formalization** (§15.3) distinguishing *record-and-classify* domains (no substrate fusion) from *operations-with-merge* domains (substrate fusion required; exercised in verifiable credentials).
4. **A confidentiality-preserving cross-reference primitive** that supports existence queries across parallel chains without content disclosure — novel to the 8th domain (bank credit intelligence).
5. **Empirical validation**: 1322/1322 unit tests passing across 8 wrapper implementations, with Bitcoin-anchored timestamps on all commits providing independent priority-date verification.

A substrate-level placement of the non-fusion and content-exclusion contracts also has cross-domain coverage implications. Each of VeilAudit [32], GeoCross [33], zkCross [34], and ForensiCross [35] supplies cross-chain confidentiality or accountability at the application or protocol layer — through linkability tags, Groth16-based verification with reputation-weighted routing, multi-protocol orchestration, or BridgeChain-mediated unanimous-agreement stage progression, respectively — so extending such mechanisms to a new application domain typically entails per-domain engineering at the cryptographic-protocol layer. By contrast, the substrate exposes non-fusion and content-excluding cross-reference contracts at the API surface, so a per-domain wrapper is concerned only with mapping the domain's records, attestations, workflow states, and terminal conditions to substrate primitives. The eight wrappers in Section 4 — five using real-world data and three using synthetic fixtures (§7.1) — were each implemented without modifying the substrate itself. We do not claim any quantified reduction in engineering effort, deployment complexity, or implementation cost; we observe only that substrate-level placement changes where domain-specific wrapper logic resides.

1.4 Roadmap

Section 2 surveys related work. Section 3 specifies the substrate. Section 4 presents the eight domain demonstrations. Section 5 introduces the confidentiality-preserving cross-reference primitive. Section 6 reports validation results. Section 7 discusses limitations, threats to validity, and ethical considerations. Section 8 concludes.

2. Related work

2.1 Distributed version control and convergent replicated data

Git [5] tracks file-content provenance via Merkle-trees of commit hashes. Two parallel branches can be **merged** into one via a three-way merge (`git merge`), producing a single resulting tree. This is appropriate for code collaboration but is precisely the property we want to *forbid* in domains where parallel authorities must remain structurally separate. Our substrate makes merge a substrate-level impossibility for record-and-classify domains (§15.3) rather than a discouraged practice.

A sharper contrast than Git is the conflict-free replicated data type (CRDT) literature [19]. CRDTs are designed so that concurrent replicas always *converge* under a well-chosen join/merge operation, using commutativity, idempotence, and monotonic semilattice structure to make conflict resolution algebraic. Our substrate makes the opposite design choice: for record-and-classify domains, **non-convergence** of independent authority histories is the intended safety property. The substrate deliberately exposes no join operation; what CRDTs treat as the central design goal, the substrate treats as a structural prohibition. Vector clocks and related distributed-causality tracking [27] are similarly oriented toward eventually-consistent reconciliation; our work is oriented toward permanent non-fusion.

2.2 Tamper-evident publication: blockchains and transparency logs

Bitcoin [6], Ethereum [7], and other public ledgers provide single-chain immutability with global consensus. Certificate Transparency [22] generalizes the underlying primitive to publicly-auditable append-only Merkle logs with inclusion and consistency proofs, focused on detecting misissued TLS certificates; Rekor [13] applies the same primitive to signed software-supply-chain metadata. OpenTimestamps [17] and RFC 3161 [28] provide the proof-of-existence layer (decentralized Bitcoin anchoring and classical PKI timestamping authorities respectively).

These systems supply *evidence layers*: they prove that a given payload was published, ordered, or existed at a point in time. Our substrate differs in three ways: (i) parallel chains are first-class, not re-encoded as smart contracts or log entries; (ii) there is no single global ordering; (iii) attestation is gauge-relative ($\sigma = \pm 1$) rather than absolute. Bitcoin OpenTimestamps and transparency logs play a *complementary* role in our system: anchors provide an external priority-date witness for substrate-internal events, but they do not decide whether two domain records share a history.

2.3 Audit logs, secure logging, and privacy-preserving audit

Audit logs (RFC 5424; SOC2 Type II requirements [8]) are append-only event streams. Schneier and Kelsey [20] introduced the canonical secure-audit-log construction with forward-integrity guarantees, targeting protection against post-compromise tampering on untrusted machines. Harpocrates [21] adds privacy: an immutable audit log that records sensitive data operations without leaking sensitive identifiers, motivated by HIPAA-style compliance regimes. Event sourcing [9] generalizes audit logs to reconstruct application state from event history.

These literatures address temporal ordering, log integrity, and log confidentiality. They do **not** address identity-by-history across multiple independent authorities, and they do not constrain which authority histories may be fused. The §5 confidentiality primitive in this paper is best understood as orthogonal to secure/private logging: production deployments would compose both (secure/private logging beneath the substrate, structural merge-absence within it).

2.4 Provenance representation and database lineage

W3C PROV [10] provides an interchange ontology for provenance graphs with entities, activities, and agents. PROV is descriptive (a graph schema) and does not address fusion semantics. Classic database provenance [23] formalizes *why*, *where*, and *how* provenance — explanations of how query outputs derive from input records via lineage. Named graphs [24] extend RDF to allow graphs themselves to be named, signed, and evaluated under task-specific trust policies.

These are representation and explanation frameworks: they describe provenance generated elsewhere. Our substrate is operational: each fold mutates state, identity is determined by history, and the structural property "no two chains can be fused" is enforced at the substrate API rather than at the descriptive layer.

2.5 Software supply-chain and operational data-lineage systems

SLSA [11], in-toto [12], and Sigstore/Rekor [13] provide build provenance for software artifacts: attestations describing where, when, and how something was produced, with optional anchoring in a transparency log. They are *artifact-centered* and typically assume one canonical builder per artifact, although they can carry multiple attestations for the same artifact.

Operational data-lineage systems for MLOps and analytics — DataHub, OpenLineage, Apache Atlas, and ML Metadata (MLMD) [29] — track how datasets and models flow through pipelines. They emphasize semantic reconciliation and aggregation across heterogeneous tools. Our SBOM demonstration (domain 3) and AI-model-lineage demonstration (domain 4) show that the substrate can record SLSA-style or pipeline-style provenance natively while preserving the structural option to register **multiple parallel builders or trainers** for the same artifact without forcing their histories to fuse. The key distinction is structural: existing systems permit semantic reconciliation across lineages; the substrate enforces lineage separation at the primitive level.

2.6 Variant interpretation in clinical genomics

ClinVar [14] aggregates germline variant interpretations from laboratories worldwide. Each submission (SCV) is preserved, but the public interface emphasizes a *consensus* view (the VCV-level review status, 1-4 stars). Our NGS domain demonstration (domain 5 of 8) shows that the same data can be modeled with **structural preservation** of each lab's chain plus orthogonal access to the ClinVar review-status integer as a separate channel.

2.7 Defensive publication strategy

Defensive publication [15, 16] is established intellectual-property practice: when a technique is novel and useful but expensive or risky to patent, organizations publish it openly to establish prior art that *prevents others* from patenting the same idea. IBM, Tesla, and Google routinely use this strategy. Our preprint serves this function for the parallel-chain provenance pattern: it is published to ensure no single party can claim exclusive rights to the underlying technique. Anchoring is two-layered: a Zenodo DOI (institutionally-mediated, INSPIRE-HEP-indexed) plus Bitcoin OpenTimestamps proofs (decentralized, independently verifiable via the public Bitcoin blockchain). We make no claim about the legal scope of defensive publication or the evidentiary admissibility of these timestamps in any specific jurisdiction; jurisdiction- specific patent-law effects and evidentiary process should be evaluated by counsel.

2.8 Privacy-preserving cross-reference and confidential workflow audit

The §5 confidentiality-preserving cross-reference primitive is adjacent to three established literatures and should be read against them. **Privacy-preserving record linkage and private set intersection** [25, 36] address the problem of matching or intersecting records across institutions without revealing the parties' raw datasets; the most direct comparator for the §5 primitive's containment-bit component is Vos et al. (2024) [36], whose privacy-preserving membership query protocol allows a centralized entity to privately query datasets held by multiple data-augmenting parties and learn only a single containment bit with party-of-origin explicitly suppressed (Vos §4.3). Private-set-intersection *cardinality* variants compute the size of the intersection without disclosing its members, which maps closely to the §5 co-lender existence-and-count primitive. **Confidential workflow auditing** (AudiWFlow [30]) models distributed workflows as directed graphs and targets accountability, non-repudiation, confidentiality, availability, and collusion detection in workflow evidence. **Privacy-preserving immutable audit logs** (Harpocrates [21]) focus on hiding sensitive identifiers in audit records while preserving log integrity.

Section 2.8 situates the confidentiality-preserving cross-reference contribution as a substrate/API-layer confidentiality contract, rather than as an application- or protocol-layer package of per-workflow cryptographic enforcement mechanisms [32–35]. In particular, we explicitly concede that the query output shape may match well-established PSI-cardinality interfaces, and we locate the novelty in the embedding of PSI-cardinality semantics inside a non-fusible multi-authority chain substrate alongside identity-by-history and regulator-scoped metadata exclusion — i.e., in the substrate-level composition and API contract rather than in PSI functionality per se. This positioning is consistent with viewing the "substrate" as a data-structural pattern implemented in software, whose technical contribution is expressed as an API-surface property rather than an application-specific proof system or encryption scheme.

To pre-empt overbroad claim overlap and to sharpen layer-correct contrasts, we now cite four representative prior systems in the order VeilAudit, GeoCross, zkCross, and ForensiCross, each of which addresses cross-chain confidentiality or accountability primarily through application/protocol-layer mechanisms or intermediary architectures rather than through a substrate-level, by-construction content-exclusion contract [32–35].

VeilAudit. VeilAudit provides a high-threat contrast because it targets cross-chain privacy-preserving accountability via auditor-testable linkability at the application layer, rather than via a substrate contract that enforces content exclusion independent of application-specific tagging [32]. Concretely, VeilAudit's confidentiality-preserving accountability is framed as "Auditor Only Linkability," using a user-generated Linkable Audit Tag that embeds a zero-knowledge proof together with a special ciphertext that only designated auditors can test for linkage [32]. The design is also explicitly two-layer, separating the transaction layer (Layer-1) from a dedicated audit chain (Layer-2), with the audit chain described as not participating in Layer-1 transaction execution or consensus formation, which keeps the audit machinery logically decoupled from transaction execution rather than fused into a single merged structure [32]. VeilAudit additionally supports threshold-controlled accountability, where identity revelation is gated by multiple independent parties (e.g., a court, regulator, or project DAO), reflecting multi-authority coordination for controlled de-anonymization under due-process conditions [32]. In contrast, the present work's defended contribution is not an application-layer deployment of linkability tags, auditor ciphertexts, or designated-auditor test procedures; rather, it is a substrate/API-layer confidentiality contract in which content non-disclosure and regulator-scoped exclusion are treated as properties of the API surface and the non-fusible substrate composition, irrespective of whether an application chooses to attach linkability tags or instantiate an auditing proof workflow [32]. The overlap with VeilAudit is therefore limited to application-layer auditability and cross-chain accountability goals, whereas the defended claim is a substrate/API-layer guarantee that enforces content-excluding cross-references by construction, without requiring per-transaction Linkable Audit Tag deployment or designated-auditor ciphertext handling [32].

GeoCross. GeoCross is a "high-threat but different layer" contrast because the excerpted design pushes confidentiality-preserving verification into Groth16 proofs plus routing/relay selection machinery, while the present work's defended claim is that content non-disclosure is an API-surface property at the substrate level rather than a property of a cryptographic exchange protocol or routing subsystem [33]. In the cited description, GeoCross targets cross-chain geological data sharing in a multi-chain Hyperledger Fabric environment and implements application/protocol-layer privacy-preserving verification using a Groth16-based zero-knowledge proof mechanism that can verify existence, integrity, and accessibility of off-chain data "without revealing the content," and reports low verification cost in terms of required bilinear pairings [33]. GeoCross also introduces a reputation-based non-interactive relay node selection protocol (RNRS) to improve cross-chain routing robustness and fairness, further underscoring that its confidentiality and authorization envelope is tied to protocol selection, relay choice, and routing mechanisms rather than to a substrate contract [33]. In contrast, the present work does not claim novelty in Groth16 verification, hierarchical encryption, or relay selection for cross-chain routing; instead, it claims a substrate/API-level primitive in which non-disclosure is enforced at the API surface and the defended novelty is the embedding of PSI-cardinality semantics in the non-fusible multi-authority chain substrate composition, not the construction of a particular ZK verification pipeline or cross-chain routing overlay [33]. The overlap with GeoCross is thus confined to application/protocol-layer privacy-preserving verification and cross-chain routing/authorization mechanisms (Groth16 and RNRS), whereas the defended claim is a substrate/API-layer content-exclusion contract that does not require applications to shoulder proof-system integration or relay-selection obligations to obtain non-disclosure semantics [33].

zkCross. zkCross is a mechanism-specific contrast that exemplifies privacy-preserving auditing achieved by orchestrating multiple cross-chain protocols in a two-layer cross-chain architecture, which is precisely the architectural style this section distinguishes from a single substrate/API contract for confidentiality-preserving cross-reference [34]. In the cited characterization, zkCross proposes a two-layer cross-chain architecture for privacy-preserving cross-chain auditing that targets Cross-chain Linkability Exposure (CLE), Incompatibility of Privacy and Auditing (IPA), and Full Auditing Inefficiency (FAI) by equipping the system with three distinct cross-chain protocols: privacy-preserving transfer, privacy-preserving exchange, and an efficient cross-chain auditing protocol [34]. The CSV-derived evidence similarly summarizes zkCross as introducing protocols built on "solid cross-chain schemes to guarantee privacy protection," reflecting a protocol-suite approach to private cross-chain auditing rather than a substrate-level API guarantee [34]. In contrast, the present work's defended contribution is not a two-layer protocol stack requiring applications to select among and orchestrate multiple privacy-preserving transfer/exchange/auditing protocols to obtain confidentiality semantics; rather, it is a substrate/API-layer contract that exposes a single surface for content-excluding cross-references, with novelty located in the substrate-level embedding of PSI-cardinality semantics and non-fusible multi-authority composition rather than in protocol multiplicity [34]. The overlap with zkCross is therefore limited to the general objective of privacy-preserving cross-chain auditing, whereas the defended claim is a substrate/API-layer confidentiality contract that avoids requiring applications to coordinate multiple cross-chain protocols to achieve content-exclusion semantics [34].

ForensiCross. ForensiCross is included as a contrast-only citation to pre-empt reviewer concerns about cross-chain forensics provenance, while maintaining the argument that such systems rely on intermediaries and agreement protocols rather than on substrate-level merge-absence and API-surface content-exclusion guarantees [35]. ForensiCross is described as a cross-chain digital-forensics collaboration design that uses a BridgeChain intermediary to facilitate exchange between heterogeneous blockchains, and its BridgeChain communication smart contract enforces stage progression only when all participating blockchains agree [35]. The CSV-derived evidence likewise emphasizes ForensiCross's architectural design around "BridgeChain" and a cross-chain/multi-chain communication protocol, and it details "Investigation Stage Progress" in which stage transactions are forwarded to the BridgeChain and stage progression requires unanimous agreement from all involved blockchains [35]. In addition, the CSV-derived evidence notes secure sharing of case information and encryption of provenance

information using the public key of the query node, reflecting confidentiality controls implemented within the forensic collaboration protocol rather than through a substrate-level content-exclusion API guarantee [35]. In contrast, we position the confidentiality-preserving cross-reference as a structural, API-layer construction that is compatible with cryptographic enforcement layers rather than a replacement for them, and that does not rely on intermediary BridgeChains or agreement-driven stage progression as the mechanism that enforces content non-disclosure [35]. The overlap with ForensiCross is thus confined to cross-chain forensics collaboration goals and protocol-mediated confidentiality controls, whereas the defended claim is a substrate/API-layer content-exclusion contract that operates without intermediary BridgeChains and without consensus-driven stage progression as the enforcement mechanism for non-disclosure semantics [35].

---

3. Substrate specification

3.1 Provenance identifiers (pids)

A `ProvenanceID` is constructed from a sequence of folds:

- **Genesis**: `genesis()` creates a fresh pid with label `'E'` and depth 1.
- **Extension**: `extend(parent)` creates a pid whose history is parent.history ++ [E_fold], incrementing depth by 1.
- **Enrichment**: `enrich(parent, polarity)` creates a pid with one additional R-fold of polarity σ ∈ {+1, −1, None}, incrementing r_count by 1.

3.2 Identity-by-history (E8)

Two pids are equal *iff* their fold-histories are identical. This is the structural primitive that prevents accidental identity collisions and enables `find` queries that distinguish “same content” from “same history.”

3.3 Gauge baseline (E26)

The first R-fold on any lineage has σ = +1 by gauge convention. This forces an interpretable baseline: the first attestation is always “asserted positive.”

3.4 Polarity (Rat B α)

Subsequent R-folds carry σ = +1 (reinforcement), −1 (revocation), or None (E27 underdetermination, used for pending review states).

3.5 Orbit decomposition (E19)

For any pid, `orbit_decomposition()` returns the structure of σ-pairs along the R-fold sequence, enabling `verify()` to detect tension (σ=+1 paired with σ=−1), all-reinforcement, and unresolved underdetermination.

3.6 Substrate-level absence of merge

The substrate exposes no primitive for fusing two pids into a third. This is a *structural* property of the API, not a policy. Application wrappers that need merge semantics (verifiable credentials, domain 2 of 8) must compensate at the application layer; the substrate exposes no method that would silently fuse parallel histories.

---

4. Eight-domain demonstration

We implemented eight wrapper modules, each mapping one domain to the substrate. All eight pass their respective unit-test suites.

This breadth — eight independent application domains, five with real-world data and three with synthetic fixtures (§7.1) — contrasts with the typical scope of application/protocol-layer cross-chain confidentiality systems, which generally instantiate against narrower anchor settings (e.g., audit-chain–based accountability in VeilAudit [32], multi-chain Hyperledger Fabric geological data sharing in GeoCross [33], two-layer cross-chain auditing in zkCross [34], and BridgeChain-mediated forensic collaboration in ForensiCross [35]) and would require per-domain protocol reengineering to span a comparable set.

4.1 Methodology

For each domain, we:

1. Identified the F1-F5 frictions (cross-lineage references, signed attestations as R-folds, workflow-state overload at σ=None, merge-analog status, terminal-state semantics).
2. Defined an app-layer composite identity key (e.g., `(variant_id, reviewer_id)` for NGS).
3. Implemented domain-specific operations using only the substrate API.
4. Wrote a test suite exercising the F1-F5 dispositions and Sections A–H structural properties.
5. Synthesized or ingested a domain-specific fixture (where public data was available; synthetic-only where regulated by law).

4.2 Summary table

#	Domain	Wrapper	Tests	F4 disposition	F1 composite key
1	Patent prosecution	`patent_prosecution_wrapper.py`	92/92	merge absent	`app_number`
2	Verifiable credentials	`vc_wrapper.py`	113/113	**merge present** (F4)	`branch_name`
3	Software bill of materials	`sbom_wrapper.py`	201/201	merge absent	`artifact_id`
4	AI model lineage (HuggingFace)	`ai_model_lineage_wrapper.py`	150/150	merge absent	`model_id`
5	NGS variant interpretation	`ngs_variant_wrapper.py`	188/188	merge absent	`(variant_id, reviewer_id)`
6	Psychiatric assessment	`psych_assessment_wrapper.py`	193/193	merge absent	`(session_id, clinician_id)`
7	Credit rating actions	`credit_rating_wrapper.py`	190/190	merge absent	`(issuer_id, agency_id)`
8	Bank credit intelligence	`bank_credit_intelligence_wrapper.py`	195/195	merge absent + **confidentiality**	`(borrower_id, bank_id)`
	Cumulative		**1322/1322**		

4.3 Friction taxonomy (F1–F5)

F1 — Substrate genesis-pid degeneracy and cross-lineage reference. Two records registered fresh both have substrate pid = genesis. The substrate cannot distinguish them; the wrapper distinguishes by the composite app-layer key. This pattern holds across all 8 domains.

F2 — Annotations as R-folds. External attestations (clinical review, rating committee decision, signed card claim) are encoded as R-folds on the relevant lineage. σ-encoding distinguishes signed (+1), revoked (−1), and pending (None).

F3 — Workflow-state overload at σ=None. σ=None alone cannot encode all workflow states (DRAFT, PENDING REVIEW, SUSPENDED, etc.). Each wrapper introduces an orthogonal app-layer status enum. Three domains (NGS, psych, credit, bank) additionally introduce a *fourth* orthogonal channel: registry-side aggregate scores (ClinVar review-status 0–4, practice-guideline level 0–3, NRSRO registered yes/no, BDDK Standart Yöntem grade 1–5).

F4 — Merge-analog status. Seven of eight domains report **merge-absent** (§15.3 inclusion criterion). The verifiable- credentials domain reports merge-present, with the wrapper recording `MergeRecord` with `is_substrate_complete=False` and compensating at app-layer.

F5 — Terminal-state. Each domain has a terminal status (WITHDRAWN, REMOVED, WRITTEN_OFF, etc.) that blocks writes; reads remain available for audit retention.

4.4 Domain-specific findings

Detailed findings for each domain are in `workshop/findings_*.md`. Notable cross-domain observations:

- The composite identity key shape is consistent across domains (string or tuple of strings). This enables uniform indexing without domain-specific schema.
- The 4th orthogonal channel (introduced in NGS) appears to be a general pattern for domains with registry-side aggregate quality scores.
- The 8th domain (banking) revealed a previously-unobserved structural property — confidentiality-preserving cross-reference — that generalizes beyond banking to any privacy-sensitive multi-authority domain.

---

5. The confidentiality-preserving cross-reference primitive

5.1 Motivation

In banking, each institution's credit intelligence on a borrower is *legally* confidential (Bankacılık Kanunu Md. 73 [3], GDPR Art. 5). Yet for systemic-risk audit, the regulator must answer questions like "how many institutions are exposed to this borrower" without seeing the content of each institution's chain.

To our knowledge, existing systems address parts of this problem — private matching (PPRL, PSI, PSI-cardinality [25]), confidential workflow auditing [30], privacy-preserving immutable audit logs [21], and provenance representation [10, 23] — but do **not** combine API-level merge-absence with multi-authority identity-by-history and regulator-scoped, content-excluding cross-reference queries in a reusable cross-domain substrate. Section 2.8 surveys these adjacent literatures; the primitive defined below is positioned as a structural API-layer construction that is **compatible with** cryptographic enforcement via these techniques rather than as a replacement for them.

5.2 Primitive

The wrapper for bank credit intelligence exposes:

```
get_co_lender_existence(borrower_id, asking_bank_id) -> (bool, int)
get_assessments_for_borrower_regulator(borrower_id, regulator) -> [metadata]
```

The first returns whether *other* banks have chains on the borrower plus the count, without revealing which banks or any content. The second returns metadata (bank_id, status, depth, r_count, bddk_risk_grade) for banks the regulator is *authorized to see*, but explicitly excludes content fields (internal grade, exposure amount, rationale, evidence).

5.3 Structural enforcement

The confidentiality contract is enforced *structurally* via API contract, not policy. The wrapper exposes no method that returns the content of another bank's chain to an asking bank. The substrate exposes no fusion primitive that

could compose chains. Section G of the test suite (24 tests) explicitly asserts that the regulator-metadata dicts do not contain content fields.

5.4 Generalization

The same primitive applies to:

- **Healthcare**: multiple specialists assessing the same patient, each chain HIPAA-private, with regulator (FDA / EMA) audit.
- **Legal**: parallel attorney-client confidential analyses of the same case, court audit possible without content exposure.
- **Tax**: multiple tax authorities assessing the same taxpayer, each jurisdiction's chain private to that authority.

This primitive is the cross-domain innovation we believe **most directly** justifies prior-art establishment for the broader substrate.

---

6. Validation

6.1 Test results

1322 unit tests pass across the 8 wrappers (1322 / 1322 = 100%). Test breakdown:

```
Patent:    92/92    (record-and-classify, F4 absent)
VC:       113/113   (operations-with-merge, F4 present)
SBOM:     201/201   (record-and-classify, F4 absent)
AI:       150/150   (record-and-classify, F4 absent)
NGS:      188/188   (record-and-classify, F4 absent)
Psych:    193/193   (record-and-classify, F4 absent)
Credit:   190/190   (record-and-classify, F4 absent)
Bank:     195/195   (record-and-classify + confidentiality)
──────────────────
Total:   1322/1322  (100%)
```

6.2 Bitcoin-anchored timestamps and DOI registration

All substrate and wrapper commits are anchored via OpenTimestamps to the Bitcoin blockchain (block confirmation pending where indicated in the public repository). Anchors establish a verifiable priority-date independent of this preprint's submission timestamp.

This preprint is additionally registered on Zenodo [18] under two published versions sharing a concept DOI 10.5281/zenodo.20127704: v1 (Rev 0, original 2026-05-11 record) at 10.5281/zenodo.20127705 and v1.1 (Rev 1, literature expansion) at 10.5281/zenodo.20128304. The DOI-bound records (CERN-operated, INSPIRE-HEP-indexed open-science repository) and the Bitcoin OpenTimestamps anchors [17] together constitute a **double-bound priority record**: an institutionally-mediated DOI plus a cryptographic, jurisdiction- independent timestamp. Both are intended to support the defensive- publication function described in §2.7 and Appendix A.

6.3 Cross-architecture caveat

All wrapper implementations and tests in this preprint were authored under a single language-model-assisted session (Claude Opus 4.7, 1M context). Cross-architecture ratification by an independent implementation in a different language-model family (per workshop convention §15.2: two-family minimum, Sonnet + Grok or Sonnet + Gemini) is required before any of the F1–F5 dispositions can be claimed as ratified rather than proposed.

---

7. Limitations and threats to validity

7.1 Synthetic fixtures

Three of the eight domains (psychiatric assessment, credit rating actions, bank credit intelligence) use synthetic fixtures because real-world data is legally restricted (HIPAA, GDPR, KVKK, banking secrecy laws). The structural properties of the wrappers are validated; real-world *distributions* are not.

7.2 No formal verification (roadmap)

The substrate is validated by unit testing (1322/1322 cases across eight wrappers), not formal verification. A natural next step is a TLA+ [31] specification of:

- **E8 (identity-by-history):** `pid_a = pid_b ⇔ history(pid_a) = history(pid_b)` as a refinement invariant.
- **E26 (gauge baseline):** the first R-fold on any lineage carries σ = +1, as a state-machine precondition.
- **E19 (orbit decomposition):** the orbit-pair function as a recursively-defined operator, with theorems on tension/reinforcement/underdetermination detection.
- **Substrate merge-absence (§3.6):** the API never exposes a primitive whose post-state has two distinct ancestor histories — a safety property.
- **§5 metadata-exclusion:** the regulator-scoped channel never returns a field outside a fixed schema, irrespective of caller — a safety property over query output.
- **F5 terminal-state write-block:** terminal status implies no subsequent state transition modifies the chain — a temporal safety property.

A Coq or Lean 4 development would additionally support proof of identity-by-history at the type level (using sized fold-sequence types as the carrier). Both directions are explicitly open work; nothing in this preprint depends on formal-verification claims.

7.3 Cryptographic confidentiality (roadmap)

The §5 confidentiality primitive is enforced by **API contract**, not by cryptography. Real-world deployment in a multi-party banking setting requires composing the substrate's structural property with one or more of the following cryptographic enforcement layers:

- **Authenticated encryption** per-authority for chain storage, so that even a compromised storage node cannot disclose chain content to peers.

- **Secure multi-party computation** [25] over chain digests, so that existence/count queries can be evaluated without any party observing another's chain identities.
- **Private set intersection** and **PSI-cardinality** [25] for the specific case of co-lender existence-and-count queries, which maps the §5 boolean-plus-count signature onto a well-studied cryptographic primitive.
- **Zero-knowledge proofs** for regulator-scoped metadata queries, so that a chain can prove conformance to a schema (e.g., BDDK Standart Yöntem 1–5 grade exists and equals N) without revealing the underlying assessment record.
- **Trusted execution environments** (Intel SGX, AMD SEV-SNP, Arm CCA) for in-bank query execution, providing hardware-level isolation from co-tenant processes and from privileged operators.

The substrate-level structural property is necessary but not sufficient for production cryptographic guarantees; this preprint deliberately establishes the structural layer first so that cryptographic constructions can be built on top of a well-defined non-fusion invariant rather than retrofitted into a system that permits fusion-by-default.

7.4 No clinical / financial-advisory claim

We make **no clinical, medical, financial-advisory, fiduciary, diagnostic, or treatment claim** for any of the domain demonstrations. The NGS, psychiatric, credit-rating, and banking implementations are *technical demonstrations* of the substrate mapping; they are not regulated medical devices, investment advisory tools, or banking software, and shall not be used as such.

7.5 Single-author work

This preprint is single-authored. Domain attributions in the header acknowledge non-technical contributors (family members and a psychologist colleague) who suggested application domains; the technical implementation and writing are the author's. Independent replication is invited and required before any patent-claim-language implication is drawn.

---

8. Conclusion

We have presented a substrate that solves the parallel-chain provenance problem across eight independent application domains with 1322/1322 passing tests. We position this preprint as **defensive prior art** for the broader pattern, complemented by Bitcoin-anchored timestamps that provide independent priority-date verification.

The most novel single contribution is the **confidentiality- preserving cross-reference primitive** introduced for bank credit intelligence (Section 5), which we believe generalizes to other privacy-sensitive multi-authority domains.

Future work includes (i) TLA+ specification of E8/E26/E19/§3.6 merge-absence/§5 metadata-exclusion/F5 write-block as safety properties, with an optional Coq or Lean 4 development at the type level (§7.2); (ii) composing the §5 structural primitive with cryptographic enforcement layers — secure multi-party computation, private set intersection / cardinality, homomorphic encryption, zero-knowledge audit, and trusted execution environments (§7.3); (iii) cross-architecture replication of all eight wrappers under the §15.2 two-family adversarial floor (Sonnet + Grok or Sonnet + Gemini); and (iv) interoperability mappings to established provenance representations (W3C

PROV [10], SLSA [11], operational data-lineage tooling [29]) so that substrate chains can emit and consume PROV-O documents and SLSA attestations without compromising the merge-absence invariant.

Reproducibility

The substrate (`provenance_engine_v2`), eight wrapper implementations, ingestion scripts, dataset summaries, and test suites are maintained in the author's GitHub repository, which is private at the time of writing and planned to be made public upon journal acceptance (and is available to reviewers on request). All repository commits are signed (SSH) and Bitcoin-anchored via OpenTimestamps under `scripts/anchors/` .

Acknowledgments

I thank my parents — my mother for identifying the NGS variant- interpretation substrate (domain 5) and my father (banker, VakıfBank) for identifying the bank credit intelligence substrate (domain 8) and challenging the over-application of the framework to direct bank sales, which led to the clarification of *defensive prior art* as the appropriate strategy and to the discovery of the confidentiality-preserving cross-reference primitive. I thank Emre Büyükasık for identifying the psychiatric assessment substrate (domain 6) and for helpful discussion of clinical workflows. I thank Yiğit Giray Şahin for early discussion of product framing.

The substrate was developed with assistance from Anthropic's Claude language model (Opus 4.7, 1M context); all design decisions, domain mappings, and writing remain the author's responsibility. Workshop pattern guards #3 (only test-verified claims), #4 (no self-ratification), #6 (no fabricated execution), #9 (same- architecture convergence is failure signature) were active throughout.

References

[1] Richards, S., et al. "Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology." *Genetics in Medicine* 17.5 (2015): 405-423.

[2] Landrum, M. J., et al. "ClinVar: improvements to accessing data." *Nucleic Acids Research* 48.D1 (2020): D835-D844.

[3] Bankacılık Kanunu (Law No. 5411), Madde 73 — Sırların saklanması. *Resmi Gazete*, 19 Ekim 2005.

[4] Wolf, T., et al. "Transformers: state-of-the-art natural language processing." *EMNLP demos* (2020).

[5] Loeliger, J., and McCullough, M. *Version Control with Git*. 2nd ed. O'Reilly Media, 2012.

[6] Nakamoto, S. "Bitcoin: a peer-to-peer electronic cash system." (2008).

[7] Buterin, V., et al. "Ethereum: a next-generation smart contract and decentralized application platform." (2014).

[8] AICPA. *Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy*. 2017.

[9] Fowler, M. "Event Sourcing." *martinfowler.com* (2005).

[10] Moreau, L., et al. “PROV-O: The PROV Ontology.” *W3C Recommendation* (2013).

[11] Google / OpenSSF. “Supply-chain Levels for Software Artifacts (SLSA).” *slsa.dev* (2022).

[12] Torres-Arias, S., et al. “in-toto: providing farm-to-table guarantees for bits and bytes.” *USENIX Security ‘19* (2019).

[13] Newman, Z., et al. “Sigstore: software signing for everybody.” *ACM CCS ‘22* (2022).

[14] Landrum, M. J., et al. “ClinVar: public archive of interpretations of clinically relevant variants.” *Nucleic Acids Research* 44.D1 (2016): D862-D868.

[15] Adams, P., and Henson-Apollonio, V. *Defensive Publishing: A Strategy for Maintaining Intellectual Property as Public Goods*. ISNAR Briefing Paper 53. 2002.

[16] IBM Technical Disclosure Bulletin archives, *IP.com* (1958–present). Defensive publication as established corporate practice.

[17] Todd, P. “OpenTimestamps: a timestamping proof standard.” *opentimestamps.org* (2016–present). Bitcoin OpenTimestamps anchors for this preprint and the underlying repository commits are available at `scripts/anchors/*.ots` in the project repository; verification: `ots verify scripts/anchors/<sha>.ots`.

[18] Karakus, B. “Parallel-Chain Provenance with Structural Merge-Absence: A Cross-Domain Substrate for Multi-Authority Audit Trails.” *Zenodo* (2026). Concept DOI (always latest version): 10.5281/zenodo.20127704. Version DOIs: v1 (Rev 0, published 2026-05-11) = 10.5281/zenodo.20127705; v1.1 (Rev 1, published 2026-05-11) = 10.5281/zenodo.20128304. Both versions share the concept DOI and are linked under a single Zenodo record family.

[19] Shapiro, M., Preguiça, N., Baquero, C., Zawirski, M. “Conflict-free replicated data types.” *13th Int. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS ‘11)* (2011): 386–400.

[20] Schneier, B., Kelsey, J. “Secure audit logs to support computer forensics.” *ACM Trans. on Information and System Security* 2.2 (1999): 159–176.

[21] “Harpocrates: privacy-preserving and immutable audit log for sensitive data operations.” *arXiv:2211.04741* (2022).

[22] Laurie, B., Langley, A., Kasper, E. “Certificate Transparency.” *RFC 6962* (2013).

[23] Cheney, J., Chiticariu, L., Tan, W.-C. “Provenance in databases: why, how, and where.” *Foundations and Trends in Databases* 1.4 (2009): 379–474.

[24] Carroll, J. J., Bizer, C., Hayes, P., Stickler, P. “Named graphs, provenance and trust.” *Proc. 14th Int. Conf. on World Wide Web (WWW ‘05)* (2005): 613–622.

[25] Pinkas, B., Schneider, T., Zohner, M. “Scalable private set intersection based on OT extension.” *ACM Trans. on Privacy and Security* 21.2 (2018): 7:1–7:35. See also Pinkas, B., Schneider, T., Tkachenko, O., Yanai, A. “Efficient circuit-based PSI with linear communication.” *EUROCRYPT ‘19* (2019). For privacy-preserving record linkage with locality-sensitive hashing and PSI cardinality, see also Vatsalan, D., Christen, P., Verykios, V. S. “A taxonomy of privacy-preserving record linkage techniques.” *Information Systems* 38.6 (2013): 946–969.

[26] Hurley, C. “Parallel provenance: what, if anything, is archival description?” *Archives and Manuscripts* 33.1 (2005): 110–145. Archival usage of “parallel provenance” denotes plural contextual narratives over records and is distinct from the data-structural usage in this paper.

[27] Lamport, L. “Time, clocks, and the ordering of events in a distributed system.” *Communications of the ACM* 21.7 (1978): 558–565.

[28] Adams, C., Cain, P., Pinkas, D., Zuccherato, R. “Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP).” *RFC 3161* (2001).

[29] Operational data-lineage and ML-metadata systems: DataHub (`datahubproject.io`), OpenLineage (`openlineage.io`), Apache Atlas (`atlas.apache.org`), ML Metadata / MLMD (`tensorflow.org/tfx/guide/mlmd`). Cited as a class representative of operational lineage tooling; no claim is made about specific implementation properties.

[30] Andrade, M., Hilbert, M., Studer, R., et al. “AudiWFlow: confidential workflow auditing for distributed systems.” *Blockchain: Research and Applications* (2022). Available via ScienceDirect, article PII S2096720922000148.

[31] Lamport, L. “Specifying systems: the TLA+ language and tools for hardware and software engineers.” Addison-Wesley (2002). See also `lamport.azurewebsites.net/tla/tla.html` .

[32] Qiao, M., Dong, H., and Gondal, I. “VeilAudit: Breaking the deadlock between privacy and accountability across blockchains.” arXiv:2510.12153 (2025).

[33] Lin, L., Feng, B., and Jing, P. “GeoCross: A privacy-preserving and fine-grained authorization scheme for cross-chain geological data sharing.” *Sensors* 25, no. 24 (2025): 7625. DOI: 10.3390/s25247625.

[34] Guo, Y., Xu, M., Cheng, X., Yu, D., Qiu, W., Qu, G., Wang, W., and Song, M. “zkCross: A novel architecture for cross-chain privacy-preserving auditing.” In *Proc. 33rd USENIX Security Symposium (USENIX Security ‘24)*, pp. 6219–6235. August 2024.

[35] Jodeiri Akbarfam, A., Dorai, G., and Maleki, H. “Secure cross-chain provenance for digital forensics collaboration.” arXiv:2406.11729 (2024).

[36] Vos, J., Pentyala, S., Golob, S., Maia, R., Kelley, D., Erkin, Z., De Cock, M., Nascimento, A. “Privacy-Preserving Membership Queries for Federated Anomaly Detection.” *Proceedings on Privacy Enhancing Technologies* 2024.3 (2024): 186–201. DOI: 10.56553/popets-2024-0074.

Appendix A — Author’s note on this preprint’s strategic intent

This preprint is published as **defensive prior art**. It is not a patent application and does not claim exclusive rights. By making this work public:

1. We establish that the parallel-chain provenance pattern with structural merge-absence (§15.3) and confidentiality-preserving cross-reference (§5) is **publicly known as of the preprint timestamp** (2026-05-11, anchored to Bitcoin block).
2. We thereby create **prior art intended to prevent any party** — including ourselves — from subsequently obtaining an exclusive patent on the broad pattern. We make no guarantee that no patent will be filed or issued; the goal is to establish the public disclosure that supports invalidation of such a claim on the broad pattern.

3. We **preserve the option** to pursue narrow patent protection for specific implementations (e.g., a particular banking implementation of §5 with additional cryptographic guarantees) without conflict, as those implementations would be substantively distinct from this preprint's public disclosure.

This is the standard defensive-publication strategy used by IBM, Tesla, Google, and other technology firms for foundational techniques where exclusive rights would be either unobtainable (35 U.S.C. §101 Alice doctrine [Alice v. CLS Bank, 2014]) or strategically suboptimal.

Preprint Rev 1 (v1.1) working draft, 2026-05-11. Comments and corrections welcome at the author's correspondence address. A revised version will be submitted to arXiv (cs.DC or cs.CR) following internal review.

End of preprint draft.